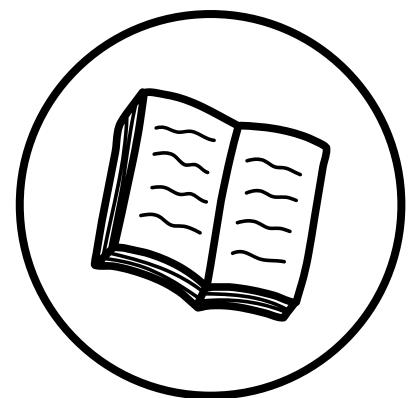


SOFTWARE ENGINEERING

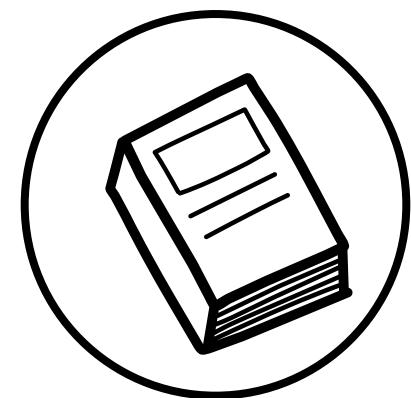
Project-1

Presented by: Team 27

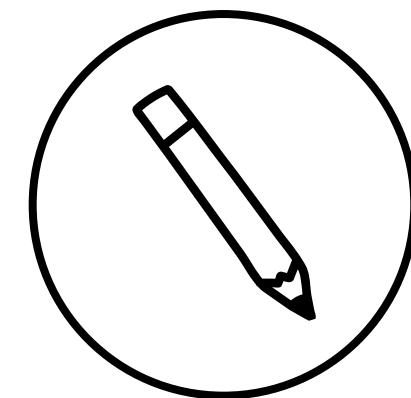
Project Objectives



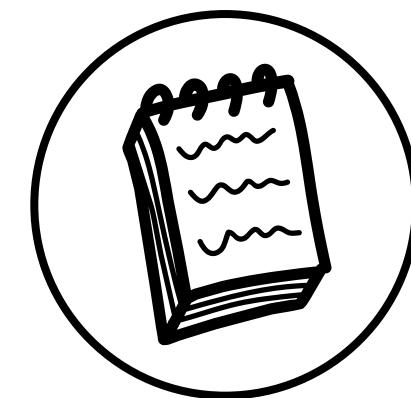
UML DIAGRAM



CODE ANALYSIS



MANUAL
REFACTORING



LLM
REFACTORING



Team Dynamics

UML DIAGRAM

VIVEK RAM
RONAK REDKAR

CODE ANALYSIS

JAYANK MAHAUR
NAITIK KARIWAL

MANUAL REFACTORING

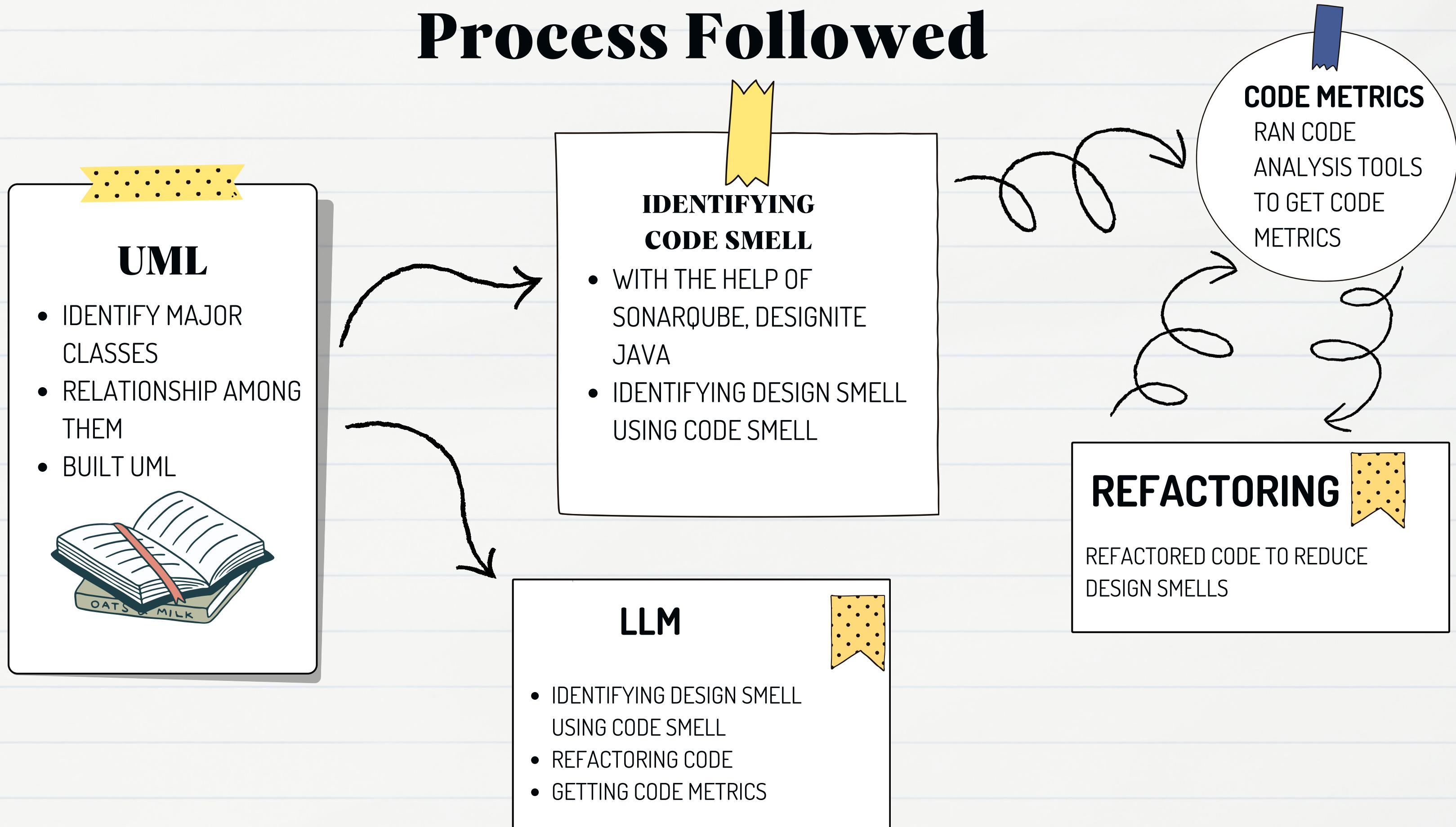
VIVEK RAM
RONAK REDKAR
NAITIK KARIWAL
JAYANK MAHAUR
KARUN CHOUDHARY

LLM REFACTORING

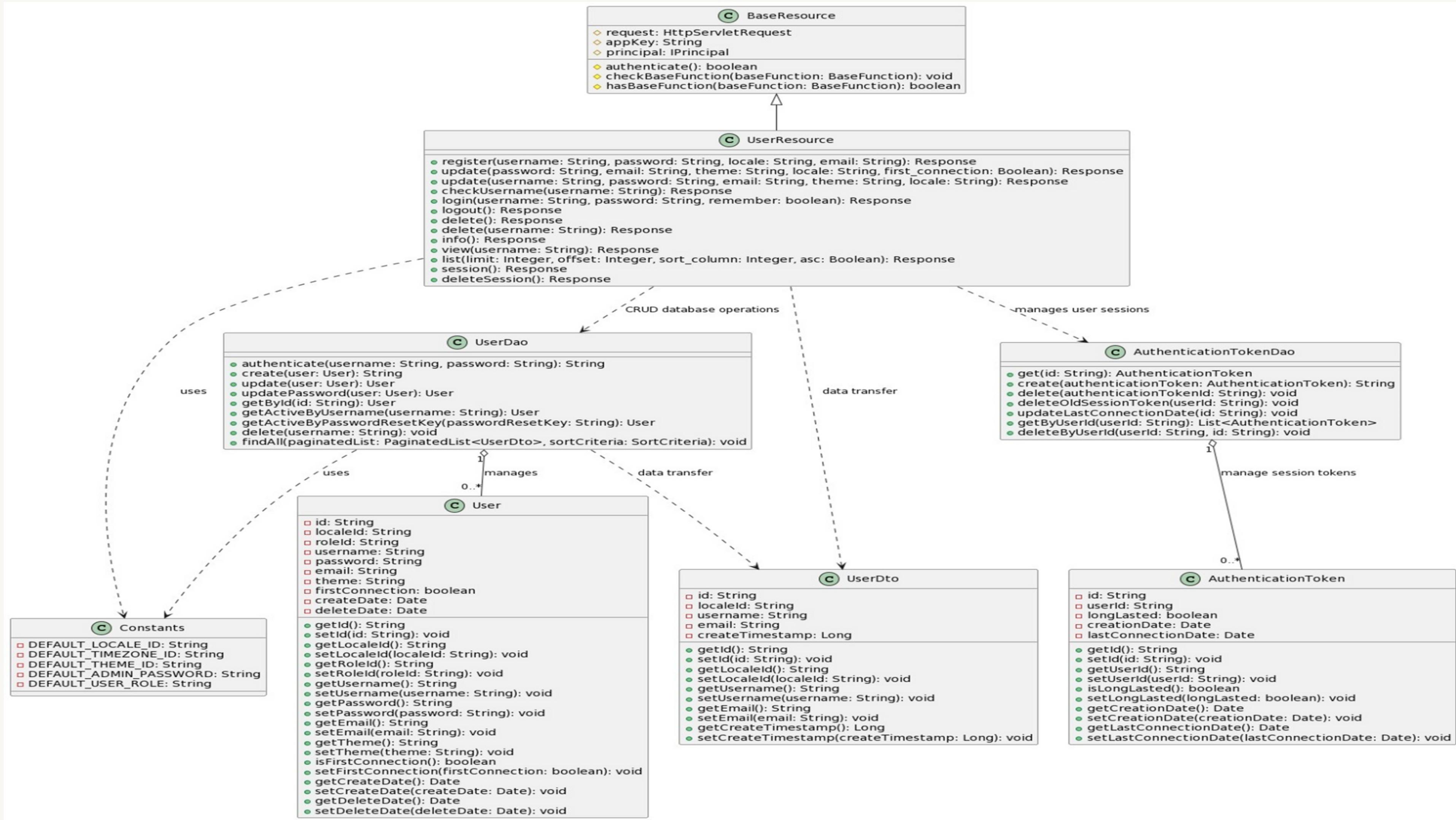
KARUN CHOUDHARY



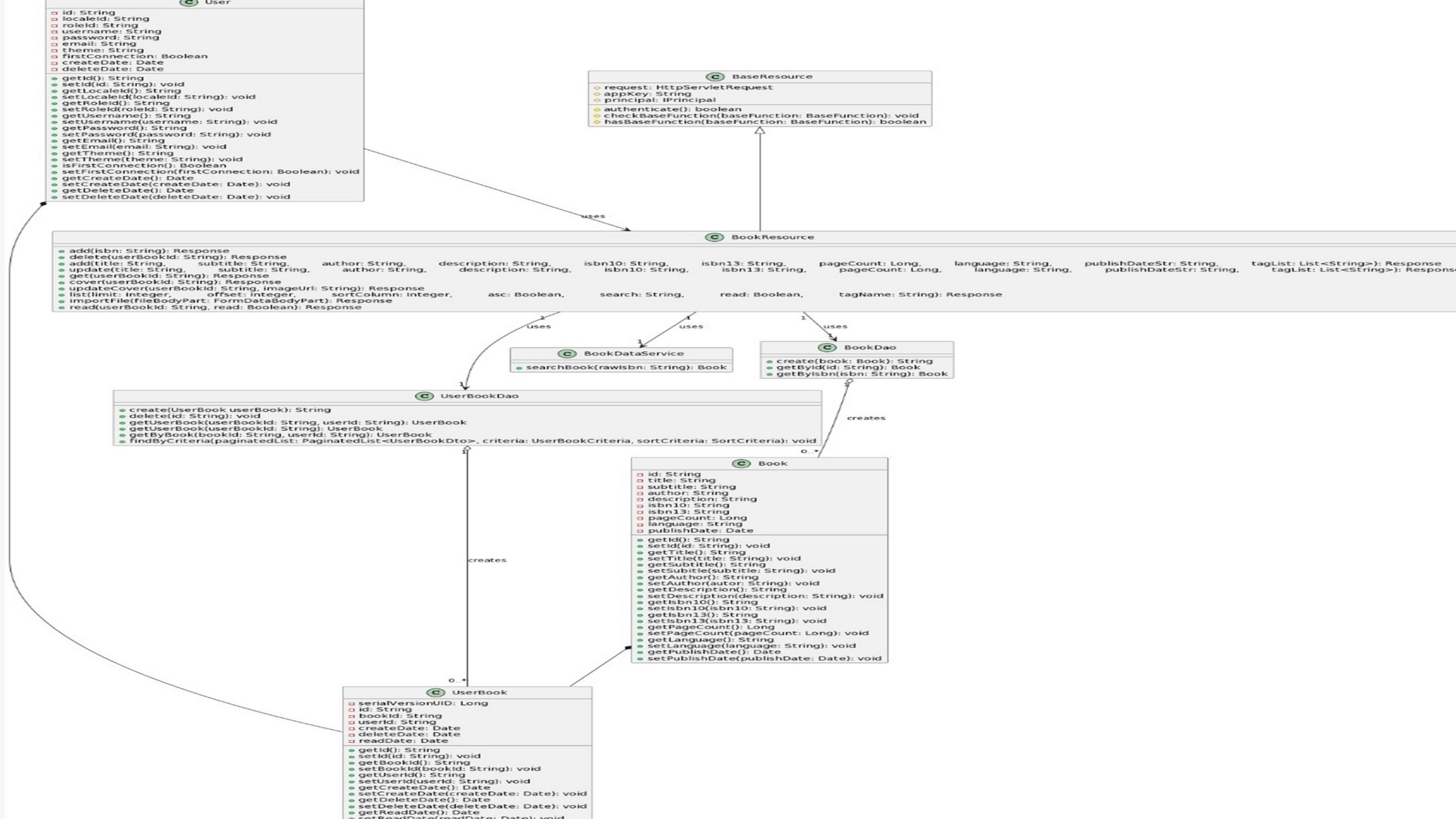
Process Followed



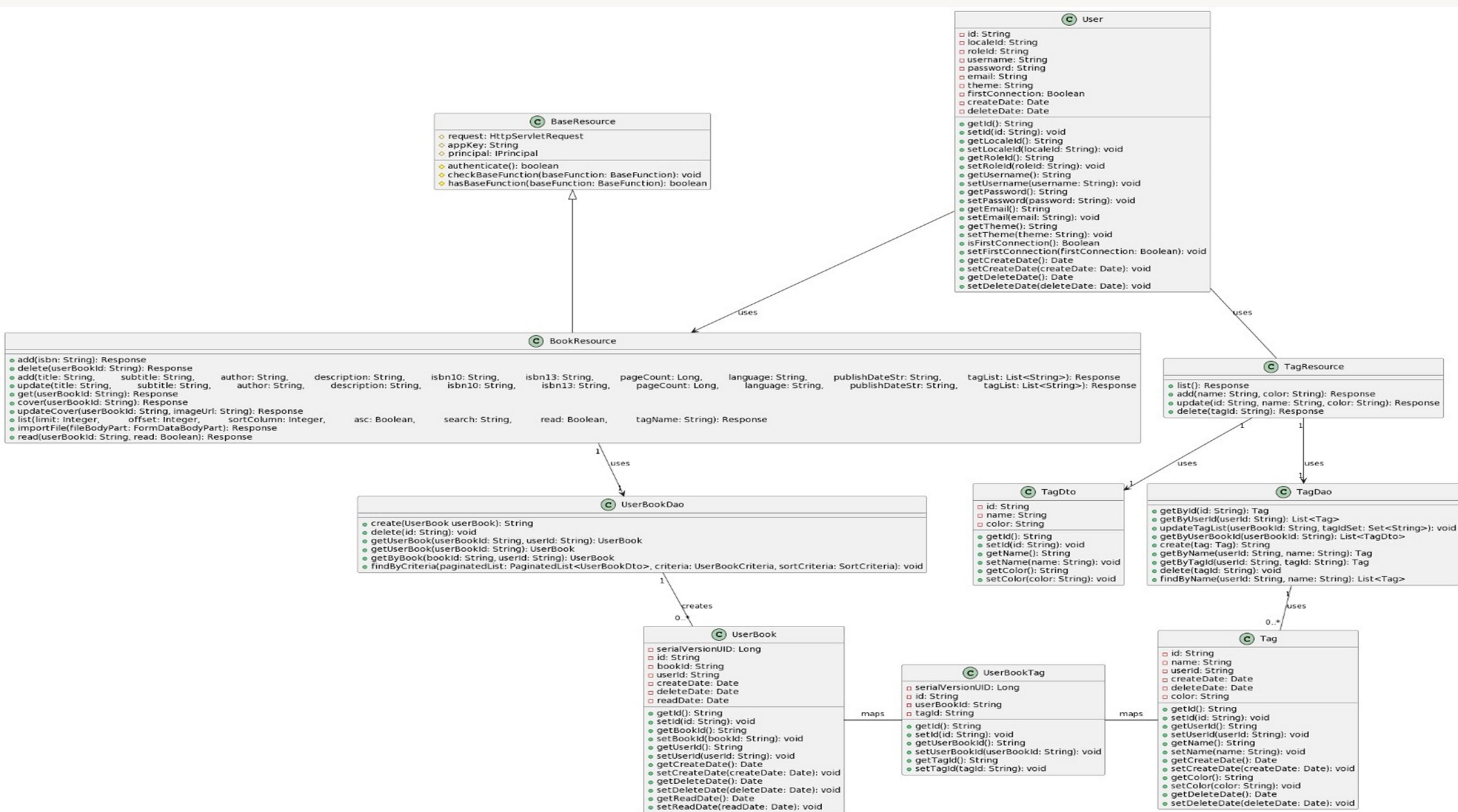
User Management System - UML Diagram



Book Addition & Display Subsystem - UML Diagram

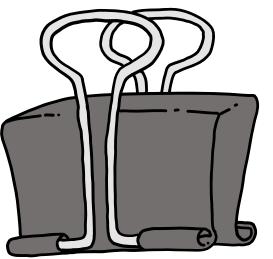


BookShelf Management Subsystem - UML Diagram



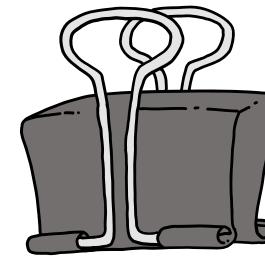
Refactoring





UserResource.java

- Code Smells: Complex Method.
- Design Smells: God Class, Insufficient Modularization.
- Refactored UserResource to adhere to the Single Responsibility Principle, extracting distinct responsibilities into separate classes.
- Created new classes – UserService, UserAuthenticationService and UserSessionService to manage User CRUD operations, authentication and session management operations respectively.



BookResource.java

- Code Smells: Complex Method.
- Design Smells: God Class, Insufficient Modularization.
- Refactor BookResource to adhere to the Single Responsibility Principle, extracting distinct responsibilities into separate classes or methods.
- Made BookFetchService.java, BookModifyService.java to break down the functionalities into separate file in order to bring modularity.
- Removed static strings from code and added them as static variables.



Constants.java

- Deficient Encapsulation.
- All constants were visible at a single place. Shifted the constants to their relevant places.
- Made Constants constructor as private so that it doesn't get instantiated.



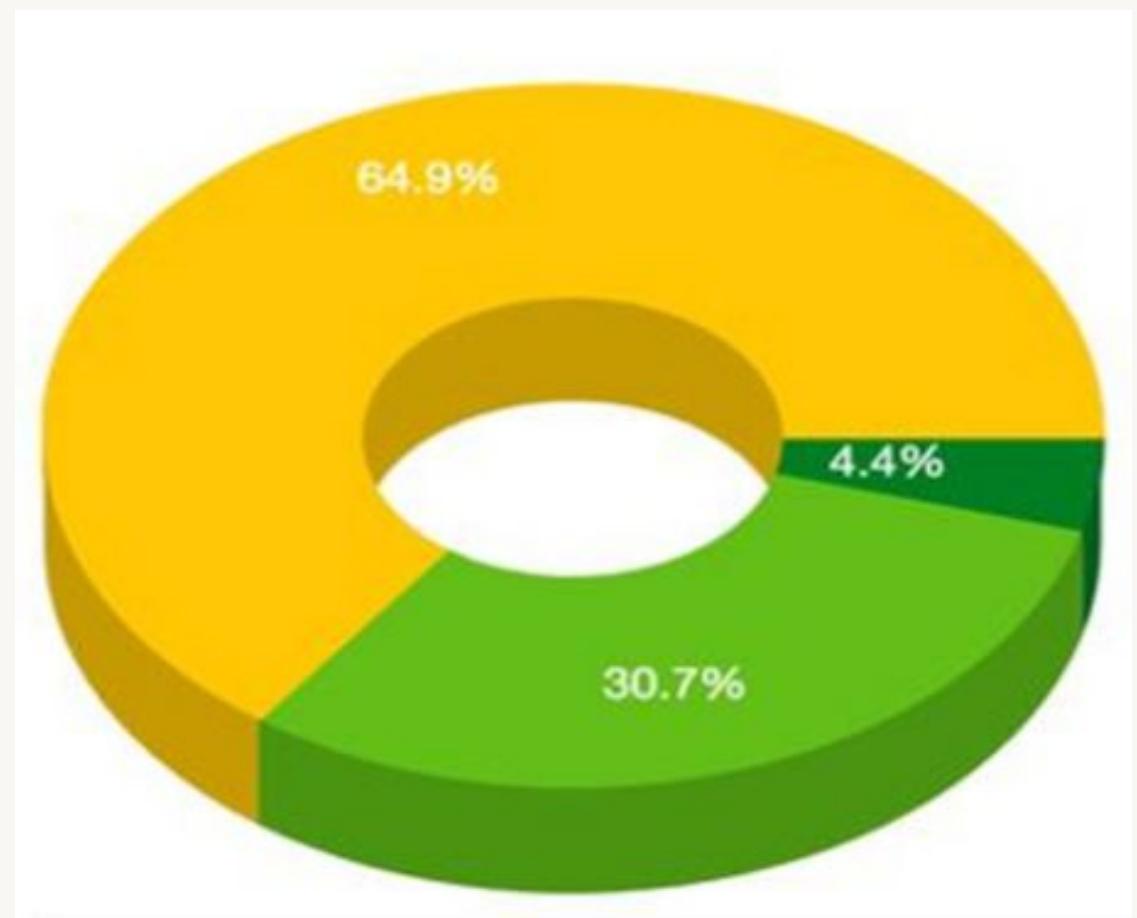
BaseResource.java

- Unutilized abstraction in BaseResource
- BaseResource was made as abstract class but which was not needed.
- Made BaseResource a normal class rather than abstract class.

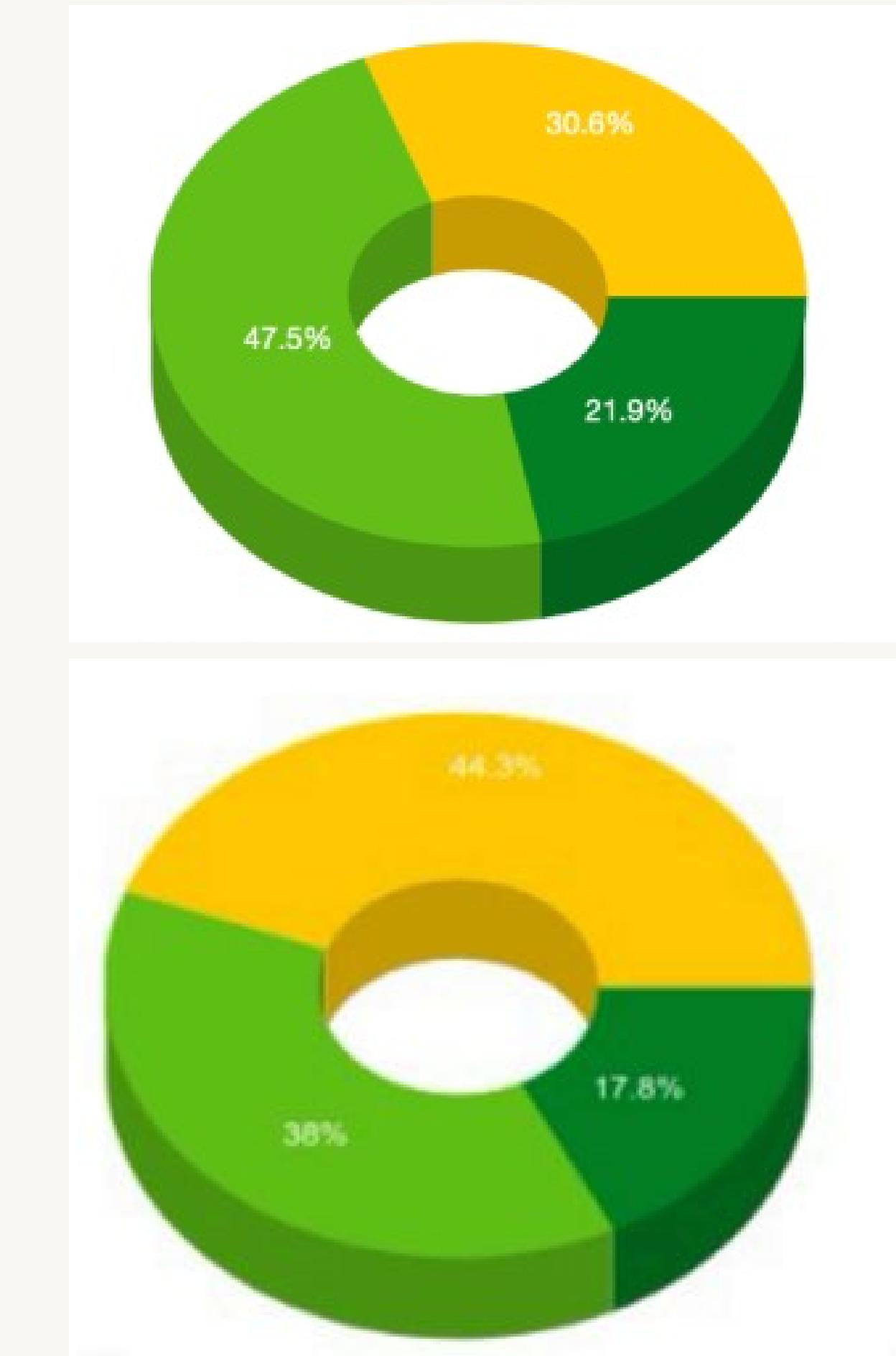
Code metrics



Complexity



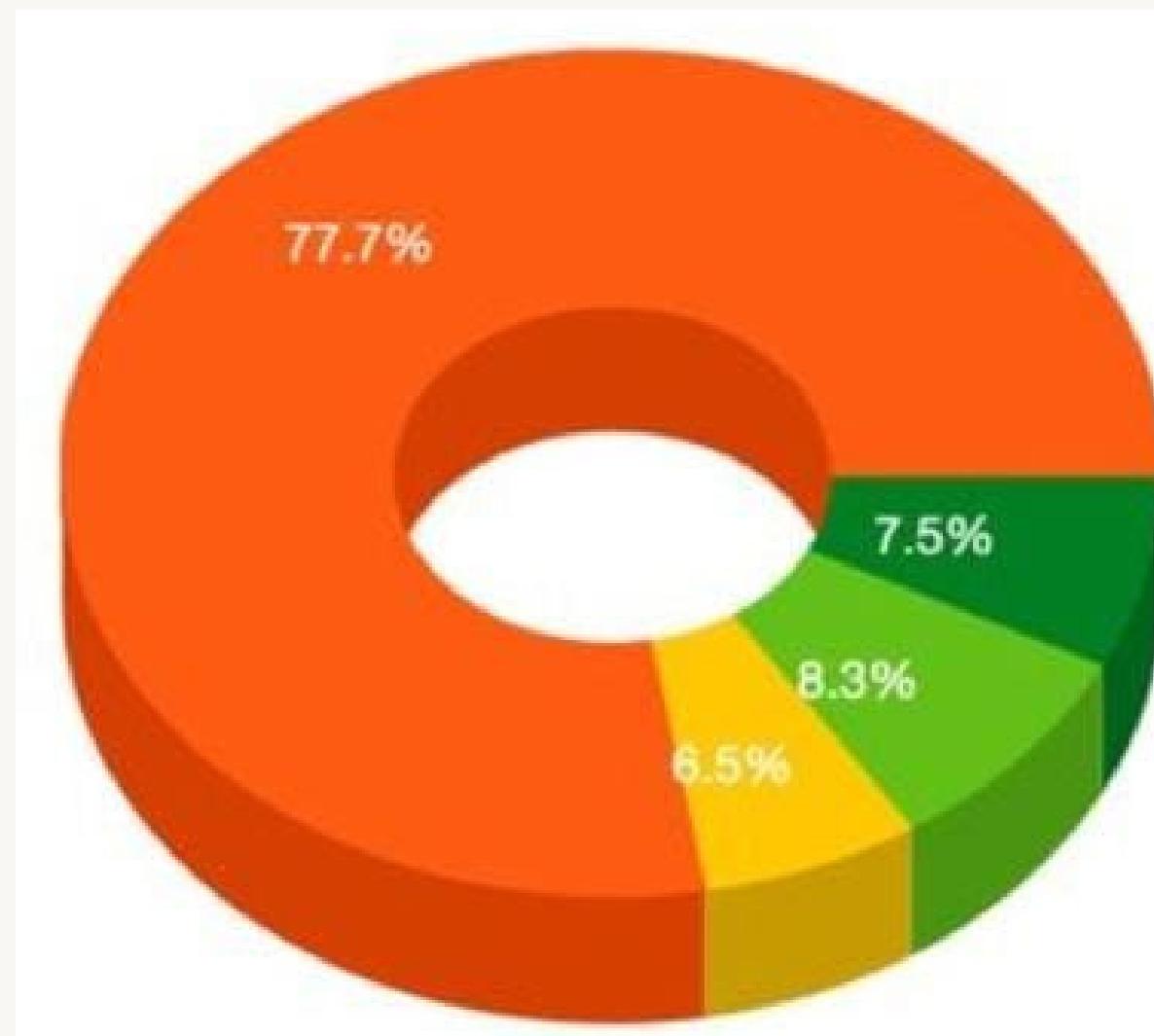
Before
Refactoring



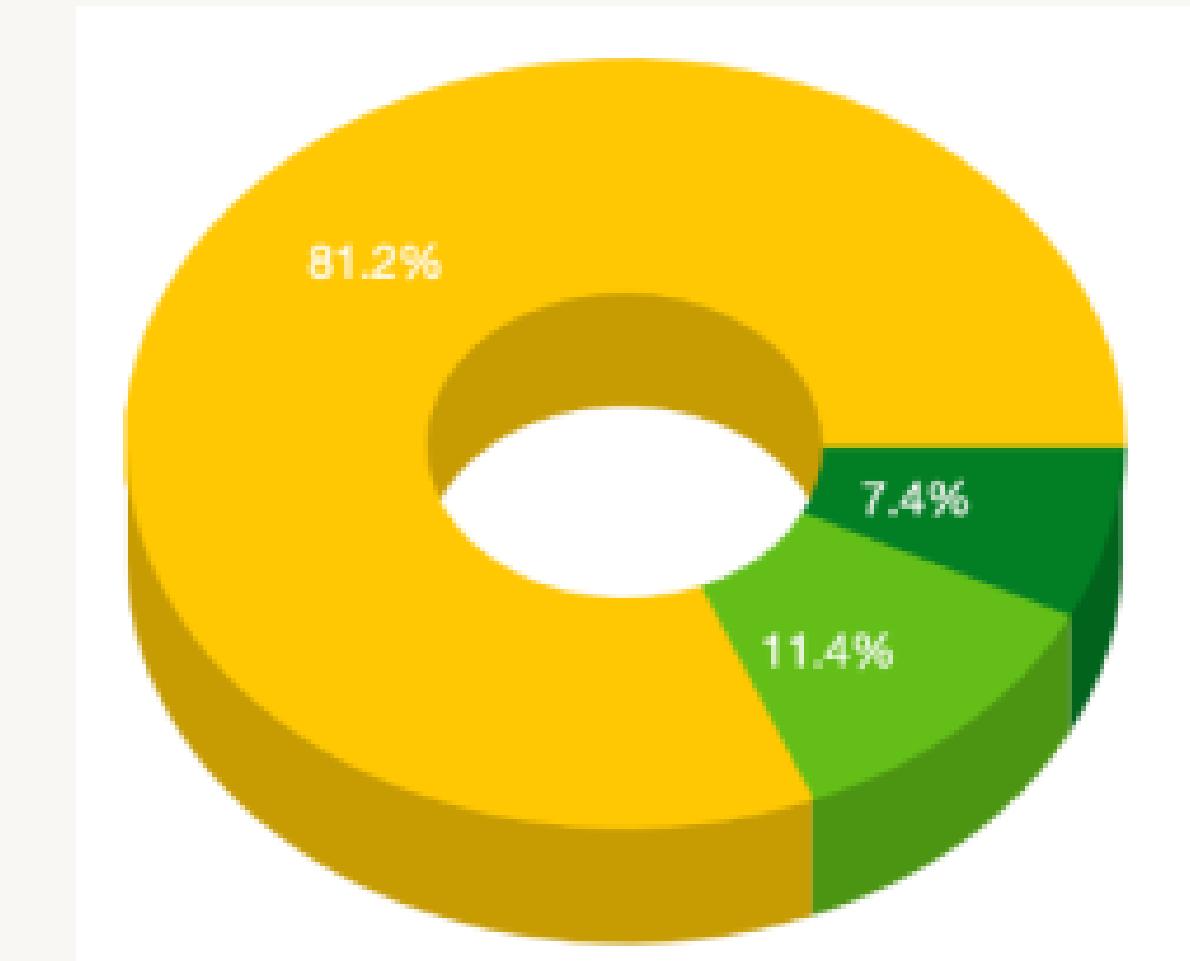
Manual
Refactored

LLM
Refactored

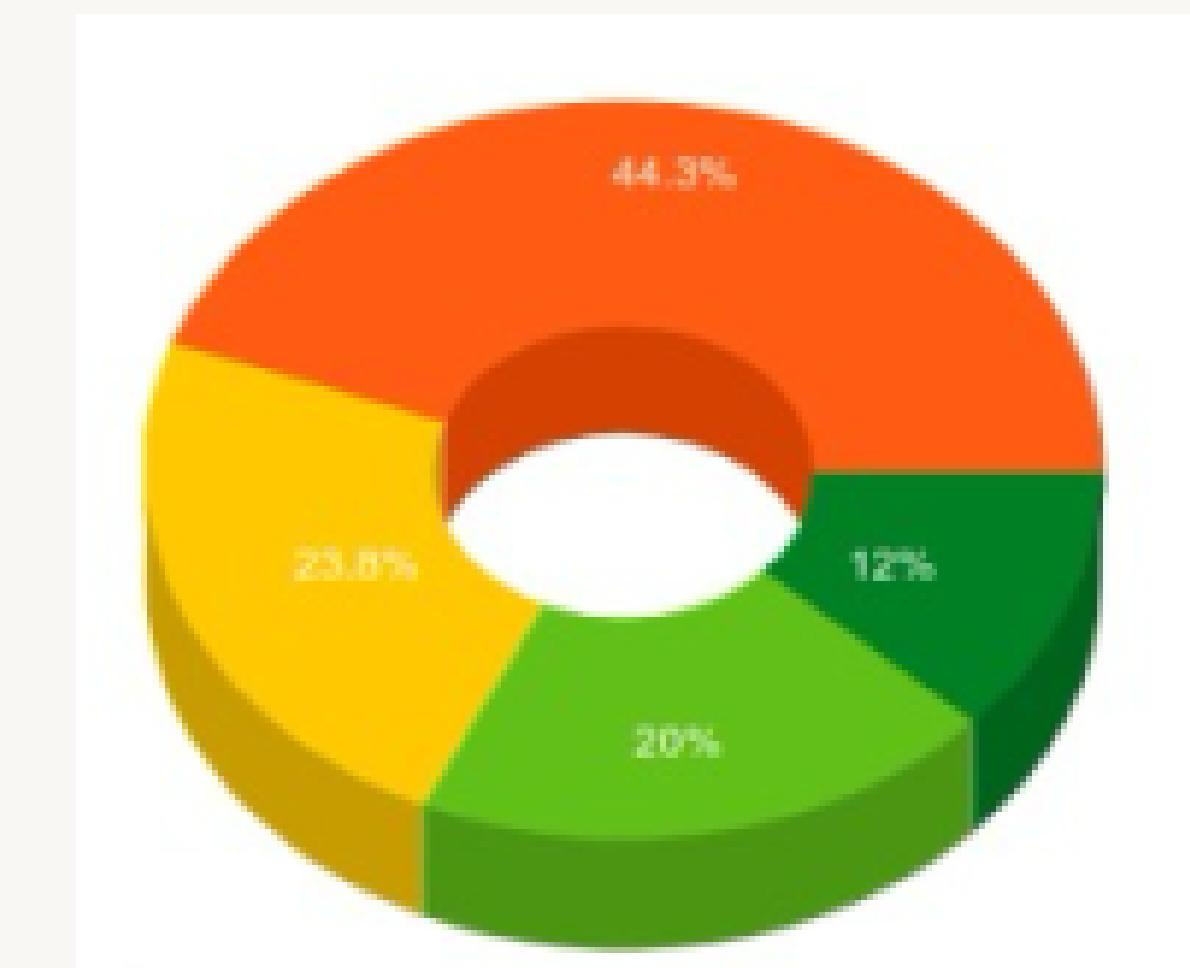
Coupling



Before
Refactoring

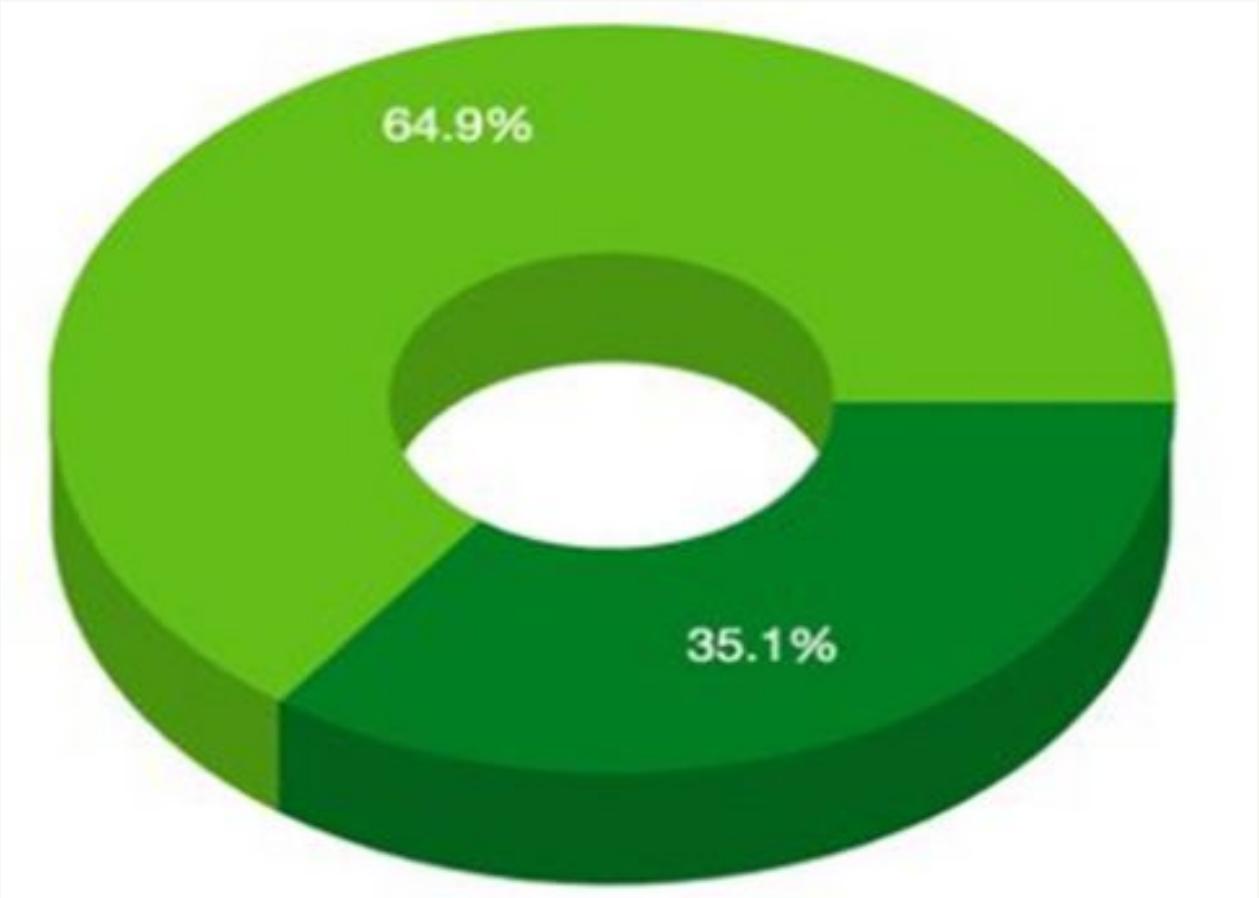


Manually
Refactored

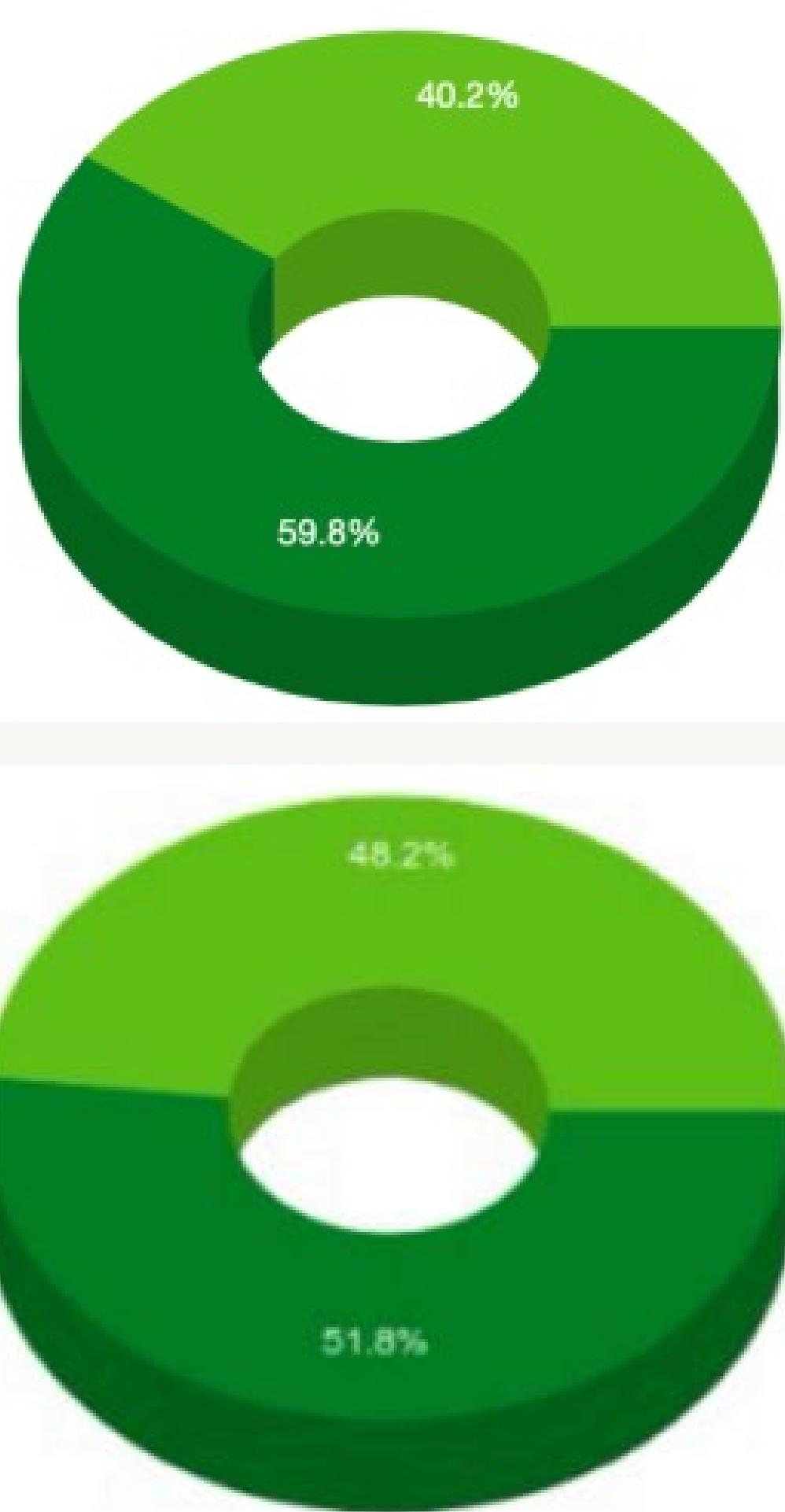


LLM
Refactored

Lack of Cohesion



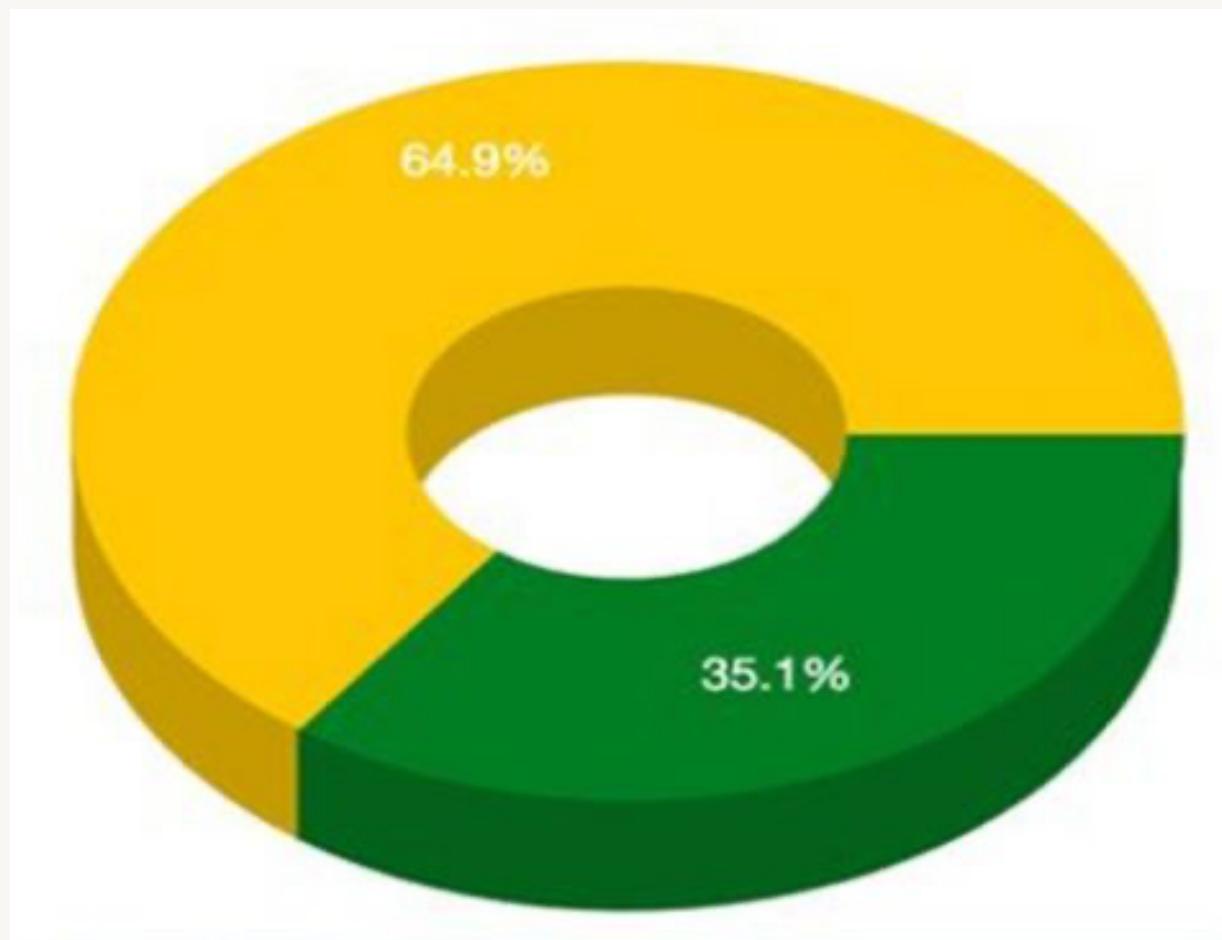
Before
Refactoring



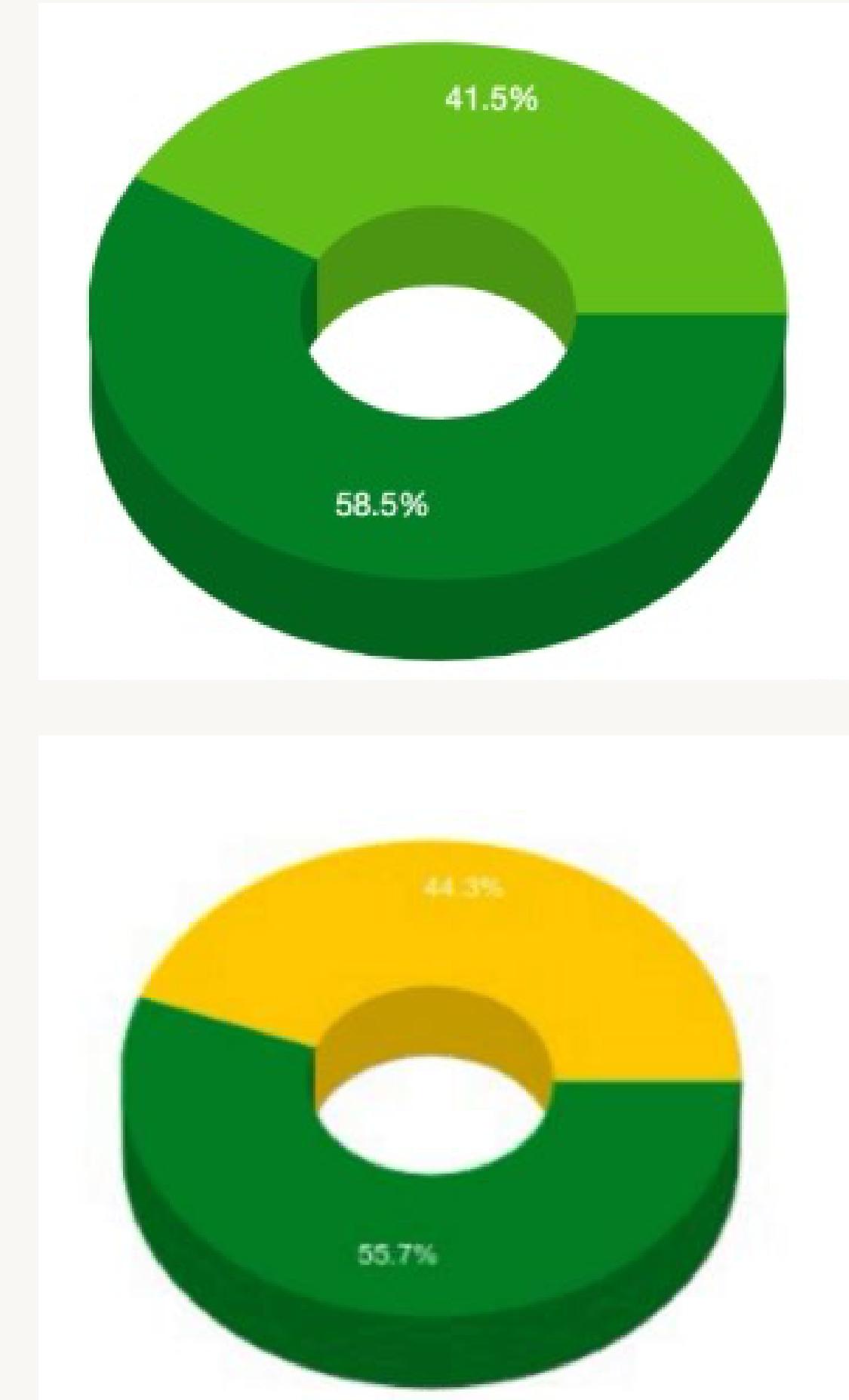
Manually
Refactored

LLM
Refactored

Weighted Method Count

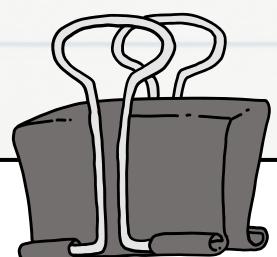


Before
Refactoring



Manually
Refactored

LLM
Refactored



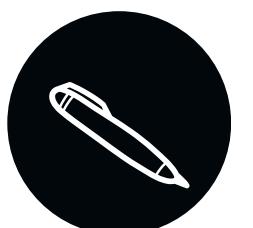
Challenges



No Proper Documentation



Compatibility issue with Sonarqube



High complexity in code.



GPT - no context of code



Thank you

